

**LAPORAN PRAKTIKUM PEKAN 8**  
**ALGORITMA DAN PEMROGRAMAN**  
**OPERATOR ARITMATIKA GUI**

Disusun oleh:

Thaariq Salam

2511532022

Dosen Pengampu: Dr. Wahyudi S.T.M.T

Asisten Praktikum: Rahmad Dwirizki Olders



**DEPARTEMEN INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS ANDALAS**  
**TAHUN 2025**

## KATA PENGANTAR

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat dan karunia-Nya, sehingga laporan praktikum mata kuliah Algoritma dan Pemrograman dengan judul “Operator Aritmatika GUI” dapat terselesaikan dengan baik.

Laporan ini disusun sebagai salah satu bentuk tugas praktikum Algoritma dan Pemrograman, dengan tujuan untuk memahami dasar-dasar pemrograman dalam bahasa Java. Pembahasan mencakup penggunaan tipe data dasar seperti *int*, *float*, *char*, *boolean*, serta penyusunan flowchart dan pseudocode dari program yang dibuat.

Penulis menyadari sepenuhnya bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis menyampaikan terima kasih yang sebesar-besarnya kepada dosen pengampu, asisten praktikum, serta rekan-rekan yang telah memberikan bimbingan dan dukungan selama penyusunan laporan ini.

Akhir kata, penulis berharap laporan ini dapat memberikan manfaat, baik bagi penulis sendiri maupun bagi para pembaca, khususnya dalam memperdalam pemahaman mengenai pemrograman dasar menggunakan bahasa Java.

Padang, 20 November 2025

Penulis

## **DAFTAR ISI**

|                              |    |
|------------------------------|----|
| KATA PENGANTAR.....          | i  |
| DAFTAR ISI .....             | ii |
| BAB I PENDAHULUAN .....      | 1  |
| 1.1    Latar Belakang .....  | 1  |
| 1.2    Tujuan Pratikum ..... | 1  |
| 1.3    Manfaat Pratikum..... | 1  |
| BAB II PEMBAHASAN .....      | 2  |
| 2.1    Kode Program .....    | 2  |
| 2.2    Langkah Kerja.....    | 4  |
| 2.3    Analisis Hasil .....  | 5  |
| BAB III PENUTUP .....        | 6  |
| 3.1    Kesimpulan.....       | 6  |
| 3.2    Saran.....            | 6  |
| DAFTAR PUSTAKA.....          | 7  |

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Praktikum adalah kegiatan belajar yang fokus pada latihan membuat kode program, mengujinya, dan menganalisis hasilnya. Tujuannya untuk menghubungkan teori yang dipelajari dengan kemampuan teknis pemrograman. Laporan ini dibuat sebagai dokumen resmi hasil praktik di mata kuliah Praktikum Algoritma dan Pemrograman dengan Java.

Laporan ini membahas hasil praktik membuat program kalkulator sederhana yang menggunakan operator aritmatika Java. Program ini dikemas dalam tampilan antarmuka grafis atau *Graphical User Interface* (GUI). Dengan memakai Java Swing, kami berlatih membuat program yang *interaktif* dan mudah digunakan.

#### **1.2 Tujuan Praktikum**

Tujuan dari praktik ini adalah:

1. Memahami langsung cara kerja operator aritmatika dasar di Java.
2. Melatih diri dalam menulis dan menjalankan kode program Java.
3. Mengembangkan kemampuan memecahkan masalah saat menerapkan logika hitungan ke dalam kode.
4. Belajar membuat tampilan program (*user interface*) menggunakan Java Swing.

#### **1.3 Manfaat Praktikum**

Manfaat dari praktik ini:

1. Menjadi media belajar yang menunjukkan bagaimana operator hitungan bekerja dalam aplikasi desktop.
2. Melatih mahasiswa membuat laporan yang sistematis dan rapi.
3. Menumbuhkan sikap teliti dan bertanggung jawab dalam pekerjaan.
4. Menjadi dokumentasi kegiatan yang berguna untuk studi selanjutnya.

## BAB II

### PEMBAHASAN

#### 2.1 Kode Program

Program ini adalah kalkulator sederhana berbasis tampilan (GUI) yang dibangun menggunakan Java Swing (*javax.swing*).

Kode Program 2.1: Class OperatorAritmatikaGUI\_2511532022

```
1 package pekan8_2511532022;
2
3 import java.awt.EventQueue;
4
5 import javax.swing.JFrame;
6 import javax.swing.JPanel;
7 import javax.swing.border.EmptyBorder;
8 import java.awt.Color;
9 import javax.swing.JLabel;
10 import javax.swing.JOptionPane;
11
12 import java.awt.Font;
13 import javax.swing.JTextField;
14 import javax.swing.JComboBox;
15 import javax.swing.DefaultComboBoxModel;
16 import javax.swing.JButton;
17 import java.awt.event.ActionListener;
18 import java.awt.event.ActionEvent;
19
20 public class OperatorAritmatikaGUI_2511532022 extends JFrame {
21
22     private static final long serialVersionUID = 1L;
23     JPanel contentPane;
24     JTextField txtBil1;
25     JTextField textBil2;
26     JTextField txtHasil;
27
28     private void pesanPeringatan(String pesan) {
29         JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.ERROR_MESSAGE);
30     }
31     private void pesanError(String pesan) {
32         JOptionPane.showMessageDialog(this, pesan, "Kesalahan", JOptionPane.ERROR_MESSAGE);
33     }
34
35     /**
36      * Launch the application.
37     */
38
39     public static void main(String[] args) {
40         EventQueue.invokeLater(new Runnable() {
41             public void run() {
42                 try {
43                     OperatorAritmatikaGUI_2511532022 frame = new OperatorAritmatikaGUI_2511532022();
44                     frame.setVisible(true);
45                 } catch (Exception e) {
46                     e.printStackTrace();
47                 }
48             }
49         });
50
51     /**
52      * Create the frame.
53     */
```

```

54④ public OperatorAritmatikaGUI_2511532022() {
55     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
56     setBounds(100, 100, 450, 300);
57     contentPane = new JPanel();
58     contentPane.setBackground(new Color(0, 128, 128));
59     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
60     setContentPane(contentPane);
61     contentPane.setLayout(null);
62
63     JLabel lblNewLabel = new JLabel("OPERATOR ARITMATIKA");
64     lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 25));
65     lblNewLabel.setBounds(72, 24, 284, 57);
66     contentPane.add(lblNewLabel);
67
68     JLabel txtBil = new JLabel("Bilangan 1");
69     txtBil.setFont(new Font("Tahoma", Font.PLAIN, 12));
70     txtBil.setBounds(24, 85, 92, 21);
71     contentPane.add(txtBil);
72
73     JTextField txtBill1 = new JTextField();
74     txtBill1.setBounds(99, 87, 96, 19);
75
76     contentPane.add(txtBill1);
77     txtBill1.setColumns(10);
78
79     JLabel txttttt = new JLabel("Bilangan 2");
80     txttttt.setFont(new Font("Tahoma", Font.PLAIN, 12));
81     txttttt.setBounds(24, 116, 92, 21);
82     contentPane.add(txttttt);
83
84     JTextField textBil2 = new JTextField();
85     textBil2.setColumns(10);
86     textBil2.setBounds(99, 116, 96, 19);
87     contentPane.add(textBil2);
88
89     JLabel cbOperator = new JLabel("Operator");
90     cbOperator.setFont(new Font("Tahoma", Font.PLAIN, 12));
91     cbOperator.setBounds(24, 150, 92, 21);
92     contentPane.add(cbOperator);
93
94     JLabel lblNewLabel_1_1_1 = new JLabel("Hasil");
95     lblNewLabel_1_1_1.setFont(new Font("Tahoma", Font.PLAIN, 12));
96     lblNewLabel_1_1_1.setBounds(24, 185, 92, 21);
97     contentPane.add(lblNewLabel_1_1_1);
98
99     JComboBox comboBox = new JComboBox();
100    comboBox.setModel(new DefaultComboBoxModel(new String[]{"+", "-", "*", "/", "%"}));
101    comboBox.setBounds(99, 147, 96, 24);
102    contentPane.add(comboBox);
103
104    JTextField txtHasil = new JTextField();
105    txtHasil.setBounds(99, 181, 96, 19);
106    contentPane.add(txtHasil);
107    txtHasil.setColumns(10);
108
109    JButton btnNewButton = new JButton("Proses");
110    btnNewButton.addActionListener(new ActionListener() {
111        public void actionPerformed(ActionEvent e) {
112            int hasil = 0;

```

```

112         if (txtBil1.getText().trim().isEmpty()) {
113             pesanPeringatan ("Bilangan 1 Harus Di Isi");
114         } else if (textBil2.getText().trim().isEmpty()) {
115             pesanPeringatan ("Bilangan 2 Harus diisi ");
116         } else {
117             {
118                 try {
119                     int a = Integer.parseInt(txtBil1.getText());
120                     int b = Integer.parseInt(textBil2.getText());
121                     int c = comboBox.getSelectedIndex();
122
123                     if (c==0) {
124                         hasil = (a+b);}
125                     if (c==1) {
126                         hasil = (a-b);}
127                     if (c==2) {
128                         hasil = (a*b);}
129                     if (c==3) {
130                         hasil = (a/b);}
131                     if (c==4) {
132                         hasil = (a%b);}
133                 }
134             catch (NumberFormatException ex) {
135                 pesanError ("Bilangan 1 dan Bilangan 2 harus diisi angka");
136             }
137             txtHasil.setText(String.valueOf(hasil));
138         }
139     }
140
141 });
142 btnNewButton.setBounds(221, 151, 85, 21);
143 contentPane.add(btnNewButton);
144
145 }

```

## 2.2 Langkah Kerja

1. Menyiapkan Tampilan Program (GUI):
  - Buat *frame* utama (*JFrame*) dan menempatkan komponen-komponen penting: dua kotak teks (*JTextField*) untuk memasukkan angka (*txtBil1* dan *textBil2*), satu kotak pilihan (*JComboBox*) berisi operator (+, -, \*, /, %), dan satu tombol *Proses*.
2. Menyiapkan Pesan Error:
  - Buat dua fungsi sederhana, *pesanPeringatan* dan *pesanError*, menggunakan *JOptionPane* agar pesan kesalahan yang muncul seragam dan jelas bagi pengguna.
3. Membuat Logika Tombol "Proses":
  - Masukkan logika utama pada tombol *Proses*. Sebelum menghitung, program akan cek: Apakah kotak Bilangan 1 atau Bilangan 2 kosong? Jika ya, munculkan pesan peringatan.

#### 4. Melakukan Perhitungan:

- Jika kedua kotak terisi, program akan mengubah teks *input* menjadi angka (*Integer.parseInt()*).
- Proses ini dilindungi dengan try-catch. Jika pengguna memasukkan huruf, eror akan ditangkap dan muncul pesan kesalahan (pesanEror).
- Operasi hitungan yang dilakukan ditentukan dari pilihan operator di combo box (indeks 0=tambah, 1=kurang, dst.).
- Hasil perhitungan ditampilkan di kotak Hasil.

### 2.3 Analisis Hasil

Program Operator Aritmatika GUI ini bekerja dengan baik dan bisa melakukan lima operasi hitungan dasar.

1. *Input* dan Operator: Pengguna bisa memasukkan angka bulat dan memilih operator yang diinginkan.
2. Validasi dan Penanganan Error:

- Program sudah bisa mendeteksi *input* kosong dan memberikan peringatan.
- Konversi tipe data *input* dari String (teks) ke *int* (angka bulat) menggunakan *Integer.parseInt()* [1] adalah langkah krusial sebelum operasi aritmatika dilakukan.
- Penggunaan blok *try-catch* untuk menangani *exception NumberFormatException* adalah cara standar Java yang memastikan program tidak rusak (*crash*) saat pengguna salah *input* [1].

Kelemahan Program: Karena semua angka yang dihitung adalah angka bulat (*int*), operasi pembagian (/) hanya menghasilkan hasil bulat (contoh: 5/2 hasilnya 2, bukan 2.5). Selain itu, program belum memiliki kode khusus untuk menangani *error* pembagian dengan angka nol.

## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Praktikum ini sukses menerapkan konsep operator aritmatika Java ke dalam aplikasi berbasis GUI (*Java Swing*). Program yang dibuat berhasil melakukan operasi hitungan dasar, dilengkapi dengan validasi input, dan penanganan kesalahan dasar, sehingga menunjukkan kemampuan dasar kami dalam membuat aplikasi yang interaktif dan stabil.

#### **3.2 Saran**

Untuk meningkatkan pemahaman konseptual dan teknis mahasiswa, disarankan agar materi praktikum selanjutnya tidak hanya berfokus pada penyalinan (*typing*) kode program saja, tetapi juga menyertakan panduan yang lebih mendalam mengenai:

1. Fungsionalitas Kode:

Penjelasan rinci mengenai fungsi spesifik dari setiap baris atau blok kode yang krusial (*critical code blocks*), misalnya apa fungsi utama dari *ActionListener* atau mengapa perlu menggunakan *Integer.parseInt()*.

2. Pemetaan Komponen dan Logika:

Penjelasan yang memisahkan secara jelas mana bagian kode yang merupakan konfigurasi komponen GUI (yang bisa diubah tampilannya, seperti *setBounds* atau *new Color*) dan mana bagian kode yang merupakan Logika Inti Program (yang wajib dipertahankan fungsinya, seperti logika *if* operator aritmatika dan blok *try-catch*).

3. Variabel Tetap dan Variabel Fleksibel:

Menandai secara eksplisit bagian-bagian kode yang bersifat esensial (tidak boleh diubah) dan bagian-bagian yang fleksibel (boleh dimodifikasi/ditingkatkan), sehingga mahasiswa terlatih untuk bereksperimen dan melakukan kustomisasi tanpa merusak fungsi dasar program.

## **DAFTAR PUSTAKA**

[1] P. Deitel dan H. Deitel, Java How to Program, Early Objects, 11th ed. Boston, MA: Pearson, 2018.